A HEURISTIC ADJACENT EXTREME POINT ALGORITHM FOR THE FIXED CHARGE PROBLEM

Warren E. Walker

RAND Corporation

Prepared for:

Public Health Service

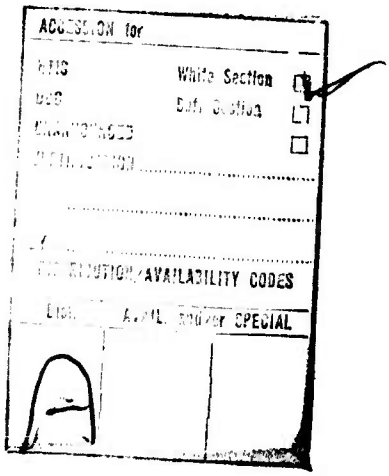
June 1973

A HEURISTIC ADJACENT EXTREME POINT ALGORITHM FOR THE FIXED CHARGE PROBLEM

Warren E. Walker

June 1973

P-5042



Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of The New York City-Rand Institute or the official opinion or policy of the City of New York. Papers are reproduced by The Rand Corporation as a courtesy to members of its staff.

A HEURISTIC ADJACENT EXTREME POINT ALGORITHM FOR THE FIXED CHARGE PROBLEM

Warren E. Walker*

The New York City-Rand Institute
545 Madison Avenue
New York, N. Y. 10022

ABSTRACT

An algorithm with three variations is presented for the approximate solution of fixed charge problems. Computational experience shows it to be extremely fast and to yield very good solutions.

The basic approach in all three variants of the algorithm is (1) to obtain a local optimum by using the simplex method with a modification of the rule for selection of the variable to enter the basic solution, and (2) once at a local optimum to search for a better extreme point by jumping over adjacent extreme points to resume iterating two or three extreme points away.

Problems in which economies of scale give rise to separable piecewise-linear concave objective functions are shown to be easily formulated as fixed charge problems.

The algorithm is currently being used by the U. S. Environmental Protection Agency's Office of Solid Waste Management Programs to solve a problem of regional solid waste planning: the selection of disposal sites to be developed and the determination of how the wastes of each municipality in a region should be distributed among the sites.

---

*Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of The New York City-Rand Institute or the official opinion or policy of the City of New York. Papers are reproduced by The Rand Corporation as a courtesy to members of its staff.

This work was supported, in part, by a grant from the U.S. Public Health Service (ES-00098).

INTRODUCTION

The fixed charge problem can be stated as

$$\left\{\begin{array}{ll} \min & z = \sum_{j=1}^{n} f_j(x_j) \\ \text{s.t.} & Ax = b \\ & x \geq 0 \\ \text{where} & f_j(x_j) = c_j x_j + k_j \delta_j \\ & \delta_j = \begin{cases} 0 \text{ if } x_j = 0 \\ 1 \text{ if } x_j > 0 \end{cases} \end{array}\right. \qquad (1)$$

This type of problem arises in many practical settings. Two of the most common of these are in warehouse or plant location, where there is a charge associated with opening the facility, and transportation problems, where there are fixed charges for transporting any goods between supply points and demand points.

If all the fixed charges k_j were zero, then problem (1) would be a linear programming problem. If some or all of the fixed charges are positive, the objective function z is concave [11] and it is well known that a concave function, defined over a convex polyhedron, takes on its minimum at an extreme point. The methods we present for the solution of (1), therefore, will examine only extreme point solutions.

The fixed charge problem can be written as a mixed-integer linear program [10, p. 253]. However, most mixed integer programming algorithms are not computationally efficient for large problems. This is true also for exact algorithms developed specifically to solve the fixed charge problem. For example, Steinberg's branch and bound algorithm [15] requires as much as 47 minutes on an IBM 360/50 to solve a 15x30 problem. Gray's decomposition approach [8] requires an average of 16 minutes to solve a 5x7 fixed charge transportation problem and as much as 22 minutes to solve a 30-site warehouse location problem on the Burroughs B-5500 [9]. Murty [13] has developed an exact algorithm which solves (1) with all $k_j=0$ to bound the total cost and then searches sys-

tematically among the extreme points adjacent to the LP optimum for the minimum total cost. Murty presents only one sample problem, solved by hand. But Gray [8] tried to solve one 6x8 and six 5x7 test problems. He was able to solve only two of the problems, running out of computer storage capacity before solving each of the other five.

Since the currently available exact algorithms generally require long computation times and large amounts of storage, a good deal of effort has been devoted to finding approximate solutions to fixed charge problems. The fixed charge transportation problem (where the A-matrix is in the form of a transportation matrix) has been investigated by Kuhn and Baumol [12] and Balinski [2]. Kuhn and Baumol suggest that an approximate solution to the problem may be obtained by forcing a highly degenerate solution. This is accomplished by making small adjustments to the right hand side (demands and supplies). The approximation is a rough one, but the computation is quite simple.

Balinski replaces the non-linear fixed charge objective function by an approximate linear objective function, and solves the resulting problem using the standard transportation algorithm. He also finds bounds on the optimal exact solution. This approach yields a rather rough approximation, and does not work well in many cases.

Cooper and Drebes [5], Steinberg [15], and Denzler [7] have developed approximate heuristic adjacent extreme point algorithms for the general fixed charge problem. Steinberg and Denzler modify the linear programming criterion for a vector to enter the basis, a technique also used in this paper. Their algorithms will be discussed more fully below. Cooper and Drebes modify the objective function at certain stages in their algorithms, and also change the criteria for vectors to enter and leave the basis. At certain times in their calculations a vector is chosen to enter the basis with the least fixed charge of the non-basic valuables. At other times a vector is chosen to leave with the highest fixed charge in the basic set.

The computational experience reported for these methods indicates that they will yield the optimal solution a high percentage of the time and, when not optimal, they provide a good approximation.

This paper will describe another adjacent extreme point algorithm (with variations) which appears to be faster than those of [5], [7] and

[15], and which yields the optimal solution a higher percentage of the time. The algorithm currently is being used on a production basis to solve large fixed charge problems [1], [4]. The algorithm will be referred to by the name SWIFT, for Simplex WIth Forcing Trials.

THE SWIFT ALGORITHM

Before proceeding with the algorithm, we make the following notational definitions:

$x_B \equiv$ vector of basis variables
$c_B \equiv$ vector of prices for x_B
$k_B \equiv$ vector of fixed charges for x_B
$a_j \equiv$ jth column of A
$B \equiv$ basis matrix
$y_j \equiv B^{-1}a_j$ (the representation of a_j in terms of B)
$z_j \equiv c_B y_j$.

The algorithm consists of two sequential phases.

Phase 1

The first phase is identical to the standard simplex procedure except that the rule for selecting the column to enter the basis is modified. In a linear program, the objective function will decrease if the entering vector, x_j, is selected so that $z_j - c_j > 0$. Because of the fixed charges, this criterion will not insure a local improvement in the fixed charge objective function. However, the non-negativity of a similar quantity involving $z_j - c_j$ and the fixed charges will produce an improvement.

Suppose that x_j is to enter the basis on the next iteration. Then, the leaving vector is x_{B_r}, where x_{B_r} is determined, as in the ordinary simplex algorithm, by

$$\theta_j = \min_k \left(\frac{x_{B_k}}{y_{kj}} , \; y_{kj} > 0 \right) = \frac{x_{B_r}}{y_{rj}} , \qquad (2)$$

and θ_j is the value which x_j assumes upon entering the basis.

If $\theta_j > 0$, as a result of such a basis change the objective function is increased by k_j, decreased by k_{B_r}, and increased or decreased by $\theta(z_j - c_j)$ depending upon the sign of $z_j - c_j$. (If $\theta_j = 0$, the objective function remains the same.) In addition, if the choice of x_{B_r} was not unique, one or more of the basic variables which were positive will become zero. In this case, even though they remain in the basis, the objective function is reduced by their fixed costs. Conversely, it is possible that, in the course of bringing x_j into the basis, some basic variables which were at a zero level will become positive. If this occurs, their fixed costs must be added in to determine the new value of the objective function. This requirement was neglected by Denzler and glossed over by Steinberg.

Let $$S = \left\{ i \left| \frac{x_{B_i}}{y_{ij}} = \frac{x_{B_r}}{y_{rj}} \right. \right\}$$

$$T = \left\{ i \left| x_{B_i} = 0,\ y_{ij} < 0 \right. \right\}.$$

Then, the entering vector, x_j, should be chosen such that

$$\Delta_j = k_j - k_{B_r} - \theta_j(z_j - c_j) - \sum_{i \in S} k_i + \sum_{i \in T} k_i < 0. \qquad (3)$$

It is possible to continue iterating using criterion (3) to choose a vector to enter the basis until $\Delta_j \geq 0$ for all non-basic columns j. However, because the objective function is concave, it is not true that when all $\Delta_j \geq 0$ a global minimum has been reached. Even though no adjacent extreme point will yield a smaller value of z, it is still possible that some other extreme point of the convex set will be better (see [12], p. 13).

This difficulty leads to phase 2 of the algorithm--a search for a better extreme point non-adjacent to the current point. Three closely related methods have been developed for phase 2. Taken together with

phase 1, which is the same for each of these methods, they constitute three heuristic algorithms for the fixed charge problem, which will be called SWIFT-1, SWIFT-2 and SWIFT-3. Aside from the treatment of degeneracy in phase 1 noted above, the phase 2 procedures distinguish the SWIFT algorithms from Steinberg's and Denzler's. The SWIFT algorithms constitute a more deliberate search for improvement and have produced better results.

Phase 2

At the end of phase 1, $\Delta_j \geq 0$ for all non-basic columns j. The phase 1 solution may or may not be the optimum. In phase 2 one or more vectors will be forced into the basis, increasing the objective function, because of the possibility that, by continuing iterations from a new point, the algorithm might move away from the old local optimum to an improved new one. That is, from a local optimum an investigation is made of nearby extreme points with larger objective values which might be adjacent to points with smaller objective values.

The three different methods for phase 2 presented below differ in (1) the number of vectors forced into the basis at a time and (2) the action taken if a forcing attempt fails to produce a better solution.

THREE SWIFT ALGORITHMS

SWIFT-1 (single forcing, non-return)

0. Find an initial feasible solution to (1).
1. Iterate with the simplex method, using criterion (3) to choose a vector to enter the basis, until $\Delta_j \geq 0$ for all non-basic columns j.
   a. Let x_0 be this phase 1 solution.
   b. Let z_0 be the corresponding value of the objective function.
2. Force a currently non-basic variable, not yet tried, into the basis, yielding a new solution x_1 with objective value $z_1 \geq z_0$. If all non-basic variables in solution x_0 have been tried without an improvement, STOP and call x_0 the (approximate) solution, otherwise go to step 3.

3. Iterate as in step 1 until $\Delta_j \geq 0$ for all non-basic columns j and a local optimum x_s is found.
   a. If $x_s \equiv x_1$ (i.e., no iterating was possible), return to solution x_0. Go to step 2.
   b. If $z_s < z_0$ a better solution has been found. Rename this solution x_0. Go to step 2.
   c. If $z_s \geq z_0$, go to step 2.

SWIFT-2 (single forcing, return)

Same as SWIFT-1 except change step 3(c) to read:

c. If $z_s \geq z_0$ return to solution x_0 (the best solution so far). Go to step 2.

SWIFT-3 (double forcing, return)

Same as SWIFT-2 except change step 2 to read:

2. Force an untried pair of non-basic variables from solution x_0 into the basis, yielding a new solution x_1 with objective value $z_1 \geq z_0$. If all pairs of non-basic variables in solution x_0 have been tried without an improvement, STOP and call x_0 the solution.

STEEPEST DESCENT

The criterion used to determine the vector to enter the basis in phase 1 could be changed to a steepest descent criterion. That is, choose vector x_j to enter if $\Delta_j < 0$ and

$$\Delta_j = \min_{i | x_i \notin B} \Delta_i$$

where Δ_i is given by (3).

The steepest descent criterion is rarely used in solving a normal linear programming problem because it involves finding θ_i for each non-basic column x_i for which $z_i - c_i > 0$. However, the algorithm described

above for the fixed charge problem requires the calculation of θ_i for at least a subset of the non-basic columns. As a result, this criterion is easy to implement and adds little time to the calculations per iteration, but reduces the number of iterations per problem by between 15 and 30 percent.

TEST RESULTS AND COMPARISONS WITH OTHER ALGORITHMS

In order to test their heuristic fixed charge algorithms, Cooper and Drebes [5] randomly generated 290-(5x10) problems with the following properties:

$$|a_{ij}| \leq 20$$

$$1 \leq c_i \leq 20$$

$$1 \leq k_i \leq 999$$

The average density of A is 50 percent.

The optimal solutions to these problems were obtained by complete enumeration. These problems and their solutions are included in Steinberg's thesis [16]. They have beer used by Cooper and Drebes, Steinberg and Denzler to test their respective algorithms.

Cooper and Drebes applied their algorithms to a set of 253 of these problems. Of these, 240 were solved optimally by their algorithm MI, and 245 by their algorithm MII. Denzler obtained optimal solutions to 169 out of 200 problems using his M-1 algorithm and all 200 using his M-3 algorithm. Steinberg obtained optimal solutions to 235 out of 250 problems using his Heuristic One algorithm, and 255 out of 268 problems using his Heuristic Two. The SWIFT algorithms were tested on the 22 problems for which Steinberg got suboptimal solutions using his algorithms. All three of the algorithms obtained optimal solutions to the 22 problems. In 16 of the 22 cases, optimality was attained by the end of phase 1.

Subsequently, the algorithms were tried on a random sample of 30 problems solved successfully by Steinberg. Again, the optimal solution was obtained for all the problems. The optimum was reached by the end of phase 1 in 26 out of the 30 problems.

The problems were solved with and without steepest descent. As an example of the effect of steepest descent, method 2 averaged a total of 18 simplex iterations/problem without and 15 iterations/problem with steepest descent. Steepest descent also reduced the average number of iterations needed to reach the optimal solution by about 3 iterations (in method 2, from 10.5 iterations to 7 iterations).

Table 1 gives the results of the SWIFT algorithms compared with those of Cooper and Drebes, Denzler and Steinberg for the 5x10 test problems.

Cooper and Drebes constructed 15x30 test problems by aggregating sets of three 5x10 problems. In these the A-matrix was formed as follows:

$$A = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{bmatrix}$$

where A_1, A_2, A_3 are 5x10 matrices. The optimal objective value for any of the 15x30 problems is the sum of the optimal objective values from the 5x10 problems associated with A_1, A_2 and A_3.

Six such 15x30 problems were constructed for SWIFT testing. SWIFT obtained optimal solutions for all six problems. Optimal phase 1 solutions were obtained in two cases. These two had A-matrices which were comprised of submatrices which had led to optimal phase 1 solutions to the constituent 5x10 problems. The other four A-matrices contained at least one submatrix from a 5x10 problem which did not produce an optimal phase 1 solution.

Table 1

COMPARISON OF ALGORITHMS
5x10 PROBLEMS

Author	Algorithm	No. tried	No. optimal	% opt.	Average no. iterations per problem	Average time per problem (sec.)	Computer[b]
Walker	SWIFT-1	52	52	100	23	5	CDC 1604
	SWIFT-1-s.d.[a]	49	49	100	17	0.5	IBM 360/65
	SWIFT-2	52	52	100	18	4	CDC 1604
	SWIFT-2-s.d.[a]	49	49	100	15	.47	IBM 360/65
	SWIFT-3	52	52	100	36	6	CDC 1604
Cooper & Drebes	MI	253	240	95	75	20	IBM 7072
	MII	253	245	97	100	20	IBM 7072
	Both	253	250	99	175	20	IBM 7072
Steinberg	Heuristic 1	250	235	94	6.8	.7	IBM 360/50
	Heuristic 2	268	255	95	15.2	1.5	IBM 360/50
Denzler	M-1	200	169	85	n.a.	1.6	IBM 7072
	M-2	200	196	98	n.a.	7.0	IBM 7072
	M-3	200	200	100	n.a.	14.0	IBM 7072

[a] s.d. ≡ steepest descent.

[b] The add times (in micro-seconds) for each of the computers used are:

IBM 7072	12.0
CDC 1604	7.2
IBM 360/50	4.0
IBM 360/65	1.3

n.a. ≡ not available.

The 15x30 problems were substantially harder to solve than the 5x10 problems. SWIFT-2 went from an average of 15 iterations/problem to an average of 86 iterations/problem. SWIFT-3 required as many as 392 iterations to solve one 15x30 problem. However, it had reached the optimal solution by iteration 35. The other iterations were spent searching for a better solution. A comparison of the results of using different algorithms for solving the 15x30 problems is given in Table 2.

Table 2

COMPARISON OF ALGORITHMS
15x30 PROBLEMS

Author	Algorithm	No. tried	No. optimal	% opt.	Average no. iterations per problem	Average time per problem (sec.)	Computer
Walker	SWIFT-2	5	5	100	86	18	CDC 1604
	SWIFT-3	2	2	100	370	60	CDC 1604
	SWIFT-3-s.d.[a]	1	1	100	376	25	IBM 360/65
Cooper & Drebes	Both MI & MII	70	63	90	1200	900	IBM 7072
Steinberg	Heuristic 1	90	75	83	12	2.16	IBM 360/50
	Heuristic 2	84	74	88	43	7.74	IBM 360/50
Denzler	M-1	22	14	64	n.a.	n.a.	
	M-3	22	17	77	n.a.	75	IBM 7072

[a] s.d. ≡ steepest descent.

n.a. ≡ not available.

Gray [9] lists 12 fixed-charge transportation problems which he solved using his exact algorithm. SWIFT-2 was applied to these same problems. Table 3 contains a comparison of Gray's computation times with those achieved by SWIFT-2. The IBM 360/65 has an add-time which is three times faster than that of the Burroughs B-5500. Even correcting for this, the SWIFT algorithm is from 2 to 835 times faster than the Gray algorithm. It failed to get the optimal solution to two problems, but in these two cases its solution was greater than the minimum cost solution by only .8 percent and 1.8 percent.

Table 3

FIXED-CHARGE TRANSPORTATION PROBLEM SOLUTIONS:
A COMPARISON OF THE GRAY AND WALKER ALGORITHMS

Problem no.	Size of A-matrix	Gray solution	Time (sec.) B-5500	Walker solution (SWIFT-2)	Time (sec.) IBM 360/65
1	7x12	329	7.7	329	1.12
1a[a]	7x12	429	7.6	429	.42
1b[b]	7x12	579	7.8	579	.43
2	10x24	202	32.6	202	1.36
3	10x24	1999	26.3	1999	1.70
4	12x32	273	171.4	273	3.69
5	12x35	245	263.8	247	3.68
6	12x35	317	146.9	317	6.11
7	12x35	1638	97.0	1668	3.01
8	12x35	2289	3262.8	2289	3.89
9	14x48	314	1510.1	314	7.61
9a[c]	14x48	2357	71.4	2357	5.89

[a]Identical to problem 1 with 20 added to each fixed charge.

[b]Identical to problem 1 with 50 added to each fixed charge.

[c]Identical to problem 9 with 250 added to each fixed charge.

THE SWIFT ALGORITHM APPLIED TO REGIONAL SOLID WASTE DISPOSAL PLANNING

In 1968 Skelly [14] developed a model for use in planning for regional solid waste disposal. The problem, viewed as a network model, was to determine the disposal facilities (sinks and trans-shipment points) which should be developed to handle the refuse generated by several communities (sources) such that the total costs of transportation, treatment and disposal are minimized. The formulation is basically a capacitated warehouse location problem. The cost function includes linear transportation costs, fixed costs for using a site and piece-wise linear concave site operating costs (which can be replaced by fixed-charge equivalents as is shown in the Appendix).

The SWIFT-2 algorithm with steepest descent was used to solve the problem using fifteen different sets of actual data. The largest problem, a 25-city, 9-site problem with 42 constraints and 282 columns (including slacks and artificials) was solved in 8.9 minutes on an IBM 360/65. Of the fifteen problems, five found improved solutions in the forcing phase, the rest reached their final solutions in phase 1.

This model has been further developed by Roy F. Westin, Inc. as part of a comprehensive state-wide solid waste management study for the New York State Department of Environmental Conservation [1] and, more recently, by the Federal Environmental Protection Agency's Office of Solid Waste Management Programs (OSWMP) for use by regional planning authorities throughout the United States. The computer program being used has a matrix generator to simplify data input, and a report generator for presenting the results in a clear and meaningful manner. But it uses the SWIFT algorithm to solve the fixed-charge problem. OSWMP is currently using the model to develop regional disposal plans for the Seattle area and for an 11-county area of Texas which includes the city of Dallas. The Texas problem has 200 refuse sources, 70 potential disposal sites and 800 source-site transportation pairs. This results in a constraint matrix having 400 rows and 870 columns (not including slack and artificial variables).

Appendix

TRANSFORMING A CONCAVE-COST LINEAR PROGRAMMING PROBLEM INTO A FIXED-CHARGE PROBLEM

Economies of scale lead to cost curves whose slope decreases as the independent variable increases. Because of the frequency of its occurrence in applications, the minimization of concave objective functions subject to linear constraints has received considerable attention in the literature [6, p. 543], [10, Chapter 3], and [3, Chapter X]. The proposed algorithms for this class of problems usually involve integer programming or lengthy search procedures.

However, it will be shown that, if the function is piece-wise linear and separable, or can be approximated by a piece-wise linear separable concave functional, the problem can be formulated as a fixed-charge problem. For this development, we first assume that any strictly concave objective function has already been approximated by a piece-wise linear separable concave functional. Then, the concave-cost linear programming problem can be stated as:

$$
\left\{
\begin{array}{lll}
\min \quad z = \sum_{j=1}^{n} \phi_j (x_j) & & \\
\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} x_j = b_i & i = 1, 2, \ldots, m & (4) \\
x_j \geq 0 & j = 1, 2, \ldots, n & \\
\text{where each of the } \phi_j (x_j) \text{ is piece-wise linear and concave.} & &
\end{array}
\right.
$$

We will construct a fixed-charge problem which is equivalent to (4). The resulting problem can then be solved by any fixed-charge algorithm.

Suppose that $\phi_j(x_j)$ is composed of r_j linear segments. Let

$c_{ij} \equiv$ slope of ith segment of $\phi_j(x_j)$:
$c_{1j} > c_{2j} > \ldots > c_{r_j j}$

$f_{ij} \equiv$ y-intercept of ith segment of $\phi_j(x_j)$ when extended to the y-axis:
$0 \leq f_{1j} < f_{2j} < \ldots < f_{r_j j}$

$h_{1j}, \ldots, h_{r_j j} \equiv$ break-points of $\phi_j(x_j)$ on the x_j axis;
$h_{0j} \equiv 0$.

Using this notation $\phi_j(x_j)$ can be represented graphically by the solid curve in Fig. 1 below.

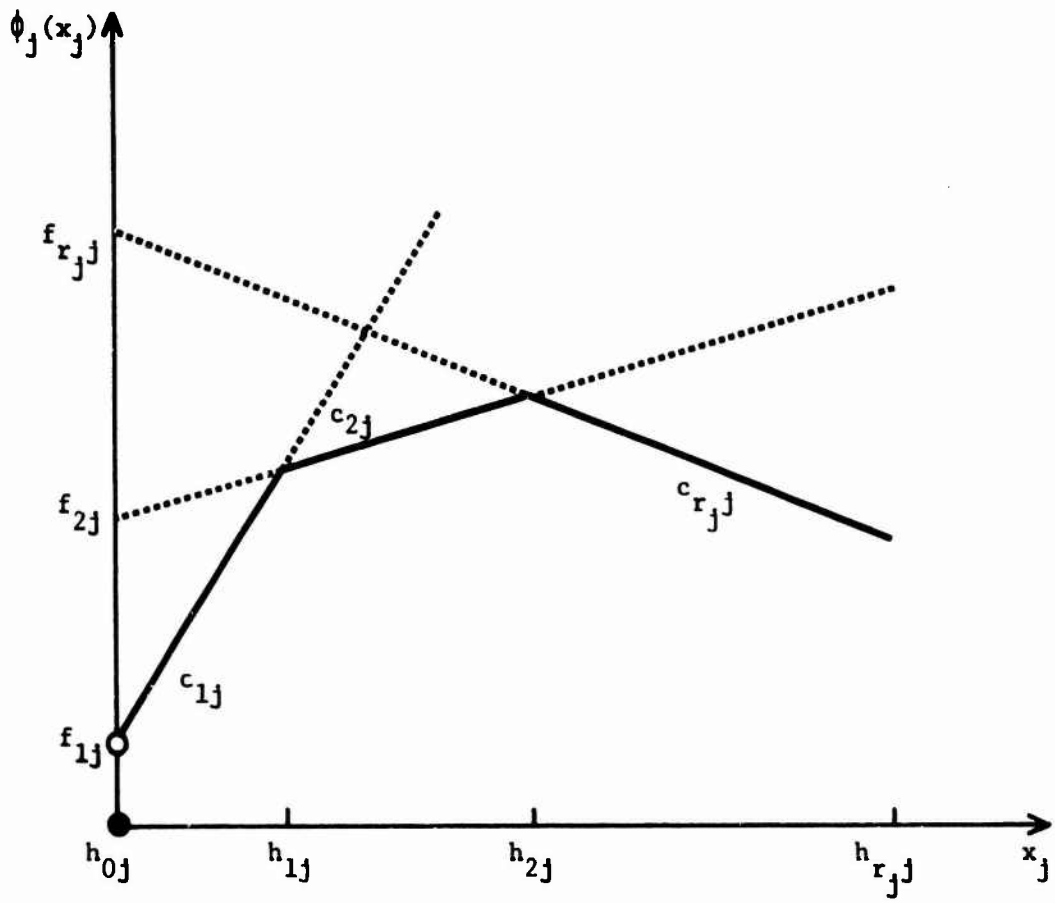


Fig. 1--Piecewise linear, concave objective function

For each activity x_j, $(j = 1, 2, \ldots, n)$, define r_j new variables $\Delta_{1j}, \Delta_{2j}, \ldots, \Delta_{r_j j}$. Associate with Δ_{ij} the variable cost c_{ij} and the fixed cost f_{ij}. Each variable Δ_{ij}, therefore, has a fixed cost objective function of the form:

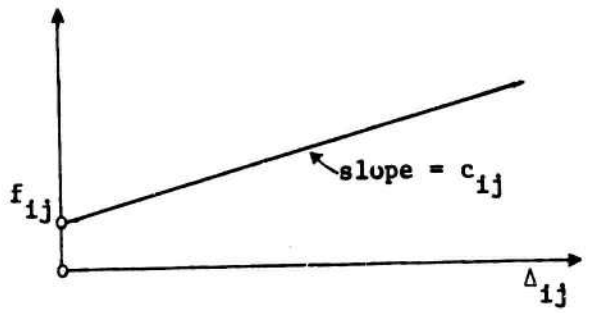


Figure 1 can be viewed as the superposition of these r_j fixed cost objective functions onto a single set of axes. The function $\phi_j(x_j)$ is the lower envelope of these cost functions. That is, for a given value of x_j, the value of the function $\phi_j(x_j)$ is

$$\min_{1 \le i \le r_j} (c_{ij}x_j + f_{ij}\delta_j)$$

$$\text{where } \delta_j = \begin{cases} 0 \text{ if } x_j = 0 \\ 1 \text{ if } x_j > 0 \end{cases}$$

Let $x_j = \Delta_{ij} + \ldots + \Delta_{r_j j}$.

Then (4) can be rewritten as:

$$\begin{cases} \min \quad z_2 = \sum_{j=1}^{n} \min_{1 \le i \le r_j} (c_{ij}\Delta_{ij} + f_{ij}\delta_{ij}) \\ \text{s.t.} \quad \sum_{j=1}^{n} a_{kj} \sum_{i=1}^{r_j} \Delta_{ij} = b_k \quad (k = 1, 2, \ldots, m) \qquad (5) \\ h_{i-1,j}\delta_{ij} \le \Delta_{ij} \le h_{ij}\delta_{ij} \\ \delta_{ij} = \begin{cases} 0 \text{ if } \Delta_{ij} = 0 \\ 1 \text{ if } \Delta_{ij} > 0. \end{cases} \end{cases}$$

The fixed-charge objective function $\sum_{j=1}^{n} \sum_{i=1}^{r_j} (c_{ij}\Delta_{ij} + f_{ij}\delta_{ij})$ could be substituted for the objective function in (5) if at most one Δ_{ij} would be positive for each j in any optimal solution.

Consider problem (5) without the constant $h_{i-1,j}\delta_{ij} \leq \Delta_{ij} \leq h_{ij}\delta_{ij}$ and with the objective changed to

$$\min \quad z = \sum_{j=1}^{n} \sum_{i=1}^{r_j} (c_{ij}\Delta_{ij} + f_{ij}\delta_{ij}).$$

The resulting problem is:

$$\begin{cases} \min \quad z = \sum_{j=1}^{n} \sum_{i=1}^{r_j} (c_{ij}\Delta_{ij} + f_{ij}\delta_{ij}) \\ \text{s.t.} \quad \sum_{j=1}^{n} a_{kj} \sum_{i=1}^{r_j} \Delta_{ij} = b_k \qquad (k = 1, 2, \ldots, m) \\ \delta_{ij} = \begin{cases} 0 \text{ if } \Delta_{ij} = 0 \\ 1 \text{ if } \Delta_{ij} > 0. \end{cases} \end{cases} \tag{6}$$

We state and prove two theorems which together show that problem (6), which is a fixed-charge problem, is equivalent to problem (4).

THEOREM 1

In an optimal solution to (6), at most one Δ_{ij} will be positive for each j.

Proof: Suppose an optimal solution has more than one Δ_{ij} positive for some j. Consider any two of them, say $\Delta_{aj} > 0$ and $\Delta_{bj} > 0$. Without loss of generality, let a < b. The cost of this solution is:

$$z_1 = K + (f_{aj} + c_{aj}\Delta_{aj} + f_{bj} + c_{bj}\Delta_{bj})$$

where $K = \sum_{p \neq j} \sum_{i=1}^{r_p} (c_{ip}\Delta_{ip} + f_{ip}\delta_{ip}) + \sum_{i \neq a,b} (c_{ij}\Delta_{ij} + f_{ij}\delta_{ij})$.

Consider reducing Δ_{aj} to 0 and increasing Δ_{bj} by Δ_{aj}, while leaving all other variables unchanged. This is also a solution, and its cost, in terms of the original variables, is:

$$z_2 = K + f_{bj} + c_{bj} (\Delta_{aj} + \Delta_{bj}).$$

Then, $$z_2 - z_1 = \Delta_{aj} (c_{bj} - c_{aj}) - f_{aj}.$$

But, by concavity, $c_{aj} > c_{bj}$, and, by assumption, $f_{aj} > 0$ and $\Delta_{aj} > 0$. Thus, $z_2 < z_1$, contradicting the assumption that we had an optimal solution.

THEOREM 2

If a Δ_{ij} is positive in an optimal solution to (6), its value will always fall between $h_{i-1,j}$ and h_{ij}.

Proof: Suppose that $\Delta_{aj} > 0$ in some solution to (6). By Theorem 1, $x_j = \Delta_{aj}$. If $h_{a-1,j} \leq \Delta_{aj} \leq h_{a,j}$, the theorem is proved, so assume that (a) $\Delta_{aj} > h_{a,j}$ or (b) $\Delta_{aj} < h_{a-1,j}$.

Case (a): $\Delta_{aj} > h_{aj}$: The objective function corresponding to this solution is

$$z_a = \sum_{s \neq j} k_s + f_{aj} + c_{aj} x_j \qquad (7)$$

where $$k_s = \sum_{i=1}^{r_s} [c_{is} \Delta_{is} + f_{is} \delta_{is}].$$

Suppose $\Delta_{a+1,j}$ were increased from 0 to x_j and Δ_{aj} were decreased from x_j to 0.
The objective function for this new solution would be

$$z_{a+1} = \sum_{s \neq j} k_s + f_{a+1,j} + c_{a+1,j}\, x_j. \qquad (8)$$

Subtracting (7) from (8) produces

$$z_{a+1} - z_a = (f_{a+1,j} - f_{aj}) + (c_{a+1,j} - c_{aj})\, x_j. \qquad (9)$$

At break-point h_{ij}, the contributions to the objective value from Δ_i and Δ_{i+1} are the same; that is

$$f_{ij} + c_{ij}\, h_{ij} = f_{i+1,j} + c_{i+1,j}\, h_{ij}$$

or

$$(f_{i+1,j} - f_{i,j}) = (c_{ij} - c_{i+1,j})\, h_{ij}. \qquad (10)$$

Substituting (10) into (9) we obtain

$$z_{a+1} - z_a = (c_{aj} - c_{a+1,j})\,(h_{aj} - x_j).$$

By concavity, $\quad c_{aj} > c_{a+1,j} \rightarrow c_{a,j} - c_{a+1,j} > 0.$

By assumption, $\quad h_{aj} < x_j \rightarrow h_{a,j} - x_j < 0.$

Thus, $z_{a+1} < z_a$, contradicting the assumption of optimality.

Case (b): $\Delta_{aj} < h_{a-1,j}$: This case is proved in a manner similar to that used for case (a) above, with $\Delta_{a-1,j}$ being increased from 0 to x_j and Δ_{aj} decreased from x_j to 0.

By repeated use of cases (a) and (b) it follows that a variable Δ_{ij} will be positive in an optimal solution only if its value falls between $h_{i-1,j}$ and h_{ij}.

ACKNOWLEDGMENT

The author wishes to thank Professor Walter Lynn, Director of the Center for Environmental Quality Management at Cornell University, for suggesting the problem, supporting the research, and contributing many helpful suggestions during the development and testing of the algorithm.

REFERENCES

1. "A Mathematical Model to Plan and Evaluate Regional Solid Waste Systems," Report by Roy F. Westin, Inc. to New York State Department of Environmental Conservation, June 1971.

2. Balinski, M. L., "Fixed Cost Transportation Problems," Naval Res. Log. Quart. 8: 41-54, 1961.

3. Charnes, A., and W. W. Cooper, Management Models and Industrial Applications of Linear Programming, Volume I, John Wiley & Sons, Inc., New York, 1961.

4. Clark, R. M., and B. P. Helms, "Decentralized Solid Waste Collection Facilities," Journal of the Sanitary Engineering Division, American Society of Civil Engineers, October 1970.

5. Cooper, L., and C. Drebes, "An Approximate Solution Method for the Fixed Charge Problem," Naval Res. Log. Quart. 8: 101-113, 1967.

6. Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963.

7. Denzler, D. R., "An Approximate Algorithm for the Fixed Charge Problem," Naval Res. Log. Quart. 16: 411-416, 1969.

8. Gray, P., "Exact Solution of the Fixed Charge Transportation Problem," Operations Research 19: 1529-1538, 1971.

9. Gray, P., "Mixed Integer Programming Algorithms for Site Selection and Other Fixed Charge Problems Having Capacity Constraints," Technical Report No. 101, Department of Operations Research, Stanford University, November 1967.

10. Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley, Reading, Massachusetts, 1964.

11. Hirsch, W. M., and G. B. Dantzig, "The Fixed Charge Problem," Naval Res. Log. Quart. 9: 413-424, 1968.

12. Kuhn, H., and W. Baumol, "An Approximate Algorithm for the Fixed Charge Transportation Problem," Naval Res. Log. Quart. 9: 1-15, 1962.

13. Murty, K. G., "Solving the Fixed Charge Transportation Problem by Ranking Extreme Points," Operations Research 16: 268-279, 1968.

14. Skelly, M. J., "Planning for Regional Refuse Systems," Ph.D. Thesis, Cornell University, September 1968.

15. Steinberg, D. I., "The Fixed Charge Problem," Naval Res. Log. Quart. 17: 217-236, 1970.

16. Steinberg, D. I., "The Fixed Charge Problem," Ph.D. Dissertation, School of Engineering and Applied Science, Washington University, St. Louis, Missouri, January 1968.